



Classification Logistic Regression

Machine Learning – CSE546
Carlos Guestrin
University of Washington

October 9, 2013

©Carlos Guestrin 2005-2014

1

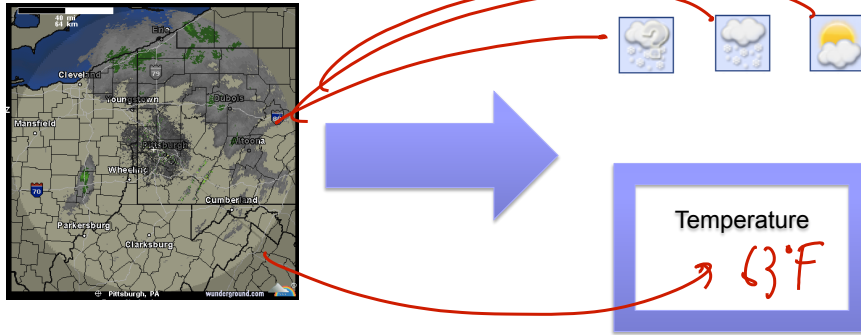


**THUS FAR, REGRESSION:
PREDICT A CONTINUOUS
VALUE GIVEN SOME INPUTS**

©Carlos Guestrin 2005-2014

2

Weather prediction revisited



©Carlos Guestrin 2005-2014

3

Reading Your Brain, Simple Example

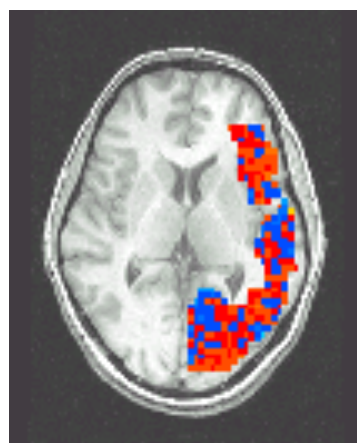
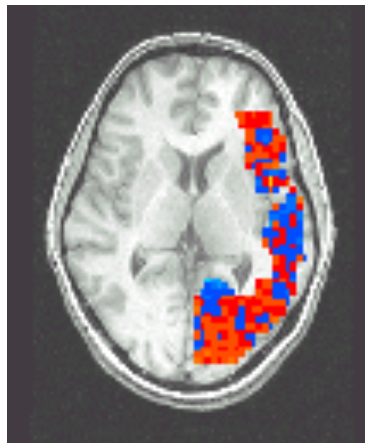
[Mitchell et al.]

Pairwise classification accuracy: 85%

Person



Animal



©Carlos Guestrin 2005-2014

4

Classification

- **Learn:** $h: \mathbf{X} \mapsto Y$
 - \mathbf{X} – features
 - Y – target classes
- Conditional probability: $P(Y|\mathbf{X})$
- Suppose you know $P(Y|\mathbf{X})$ exactly, how should you classify?
 - Bayes optimal classifier:
- **How do we estimate $P(Y|\mathbf{X})$?**

©Carlos Guestrin 2005-2014

5

Link Functions

- Estimating $P(Y|\mathbf{X})$: Why not use standard linear regression?
- Combining regression and probability?
 - Need a mapping from real values to $[0,1]$
 - A link function!

©Carlos Guestrin 2005-2014

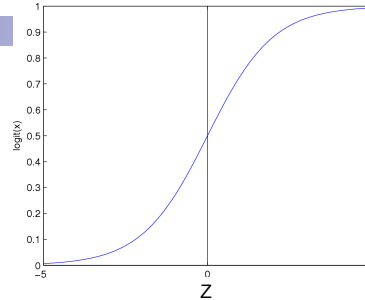
6

Logistic Regression

Logistic function (or Sigmoid): $\frac{1}{1 + \exp(-z)}$

- Learn $P(Y|\mathbf{X})$ directly
 - Assume a particular functional form for link function
 - Sigmoid applied to a linear function of the input features:

$$P(Y = 0|X, W) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)}$$



Features can be discrete or continuous!

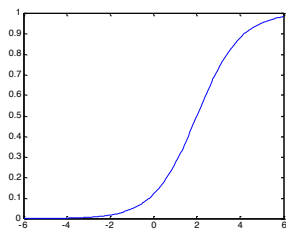
©Carlos Guestrin 2005-2014

7

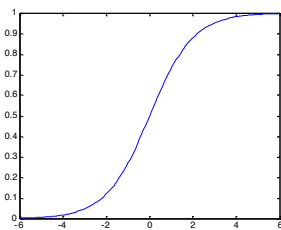
Understanding the sigmoid

$$g(w_0 + \sum_i w_i x_i) = \frac{1}{1 + e^{w_0 + \sum_i w_i x_i}}$$

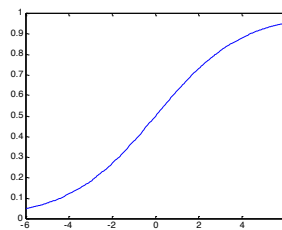
$w_0 = -2, w_1 = -1$



$w_0 = 0, w_1 = -1$



$w_0 = 0, w_1 = -0.5$

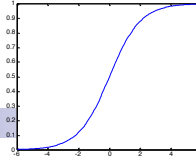


©Carlos Guestrin 2005-2014

8

Logistic Regression – a Linear classifier

$$\frac{1}{1 + \exp(-z)}$$



$$g(w_0 + \sum_i w_i x_i) = \frac{1}{1 + e^{w_0 + \sum_i w_i x_i}}$$

©Carlos Guestrin 2005-2014

9

Very convenient!

$$P(Y = 0 | X = \langle X_1, \dots, X_n \rangle) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)}$$

implies

$$P(Y = 1 | X = \langle X_1, \dots, X_n \rangle) = \frac{\exp(w_0 + \sum_i w_i X_i)}{1 + \exp(w_0 + \sum_i w_i X_i)}$$

implies

$$\frac{P(Y = 1 | X)}{P(Y = 0 | X)} = \exp(w_0 + \sum_i w_i X_i)$$

implies

$$\ln \frac{P(Y = 1 | X)}{P(Y = 0 | X)} = w_0 + \sum_i w_i X_i$$

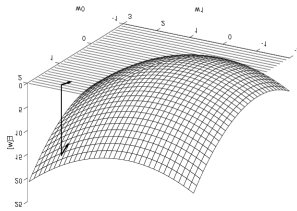
linear
classification
rule!

©Carlos Guestrin 2005-2014

10

Optimizing concave function – Gradient ascent

- Conditional likelihood for Logistic Regression is concave. Find optimum with gradient ascent



Gradient: $\nabla_{\mathbf{w}} l(\mathbf{w}) = \left[\frac{\partial l(\mathbf{w})}{\partial w_0}, \dots, \frac{\partial l(\mathbf{w})}{\partial w_n} \right]^T$

Step size, $\eta > 0$

Update rule: $\Delta \mathbf{w} = \eta \nabla_{\mathbf{w}} l(\mathbf{w})$

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \frac{\partial l(\mathbf{w})}{\partial w_i}$$

- Gradient ascent is simplest of optimization approaches
 - e.g., Conjugate gradient ascent can be much better

Loss function: Conditional Likelihood

- Have a bunch of iid data of the form:

- Discriminative (logistic regression) loss function:
Conditional Data Likelihood

$$\ln P(\mathcal{D}_Y | \mathcal{D}_X, \mathbf{w}) = \sum_{j=1}^N \ln P(y^j | \mathbf{x}^j, \mathbf{w})$$

Expressing Conditional Log Likelihood

$$P(Y = 0|X, \mathbf{w}) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)}$$
$$l(\mathbf{w}) \equiv \sum_j \ln P(y^j | \mathbf{x}^j, \mathbf{w})$$
$$P(Y = 1|X, \mathbf{w}) = \frac{\exp(w_0 + \sum_i w_i X_i)}{1 + \exp(w_0 + \sum_i w_i X_i)}$$

$$\ell(\mathbf{w}) = \sum_j y^j \ln P(Y = 1 | \mathbf{x}^j, \mathbf{w}) + (1 - y^j) \ln P(Y = 0 | \mathbf{x}^j, \mathbf{w})$$

©Carlos Guestrin 2005-2014

13

Maximizing Conditional Log Likelihood

$$P(Y = 0|X, W) = \frac{1}{1 + \exp(w_0 + \sum_i w_i X_i)}$$
$$P(Y = 1|X, W) = \frac{\exp(w_0 + \sum_i w_i X_i)}{1 + \exp(w_0 + \sum_i w_i X_i)}$$
$$l(\mathbf{w}) \equiv \ln \prod_j P(y^j | \mathbf{x}^j, \mathbf{w})$$
$$= \sum_j y^j (w_0 + \sum_i w_i x_i^j) - \ln(1 + \exp(w_0 + \sum_i w_i x_i^j))$$

Good news: $l(\mathbf{w})$ is concave function of \mathbf{w} , no local optima problems

Bad news: no closed-form solution to maximize $l(\mathbf{w})$

Good news: concave functions easy to optimize

©Carlos Guestrin 2005-2014

14

Maximize Conditional Log Likelihood: Gradient ascent

$$l(\mathbf{w}) = \sum_j y^j (w_0 + \sum_i^n w_i x_i^j) - \ln(1 + \exp(w_0 + \sum_i^n w_i x_i^j))$$

Gradient Ascent for LR

Gradient ascent algorithm: iterate until change $< \epsilon$

$$w_0^{(t+1)} \leftarrow w_0^{(t)} + \eta \sum_j [y^j - \hat{P}(Y^j = 1 | \mathbf{x}^j, \mathbf{w}^{(t)})]$$

For $i=1, \dots, k$,

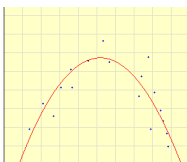
$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 | \mathbf{x}^j, \mathbf{w}^{(t)})]$$

repeat

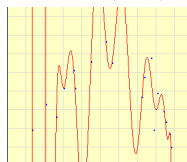
Regularization in linear regression

- Overfitting usually leads to very large parameter choices, e.g.:

$$-2.2 + 3.1 X - 0.30 X^2$$



$$-1.1 + 4,700,910.7 X - 8,585,638.4 X^2 + \dots$$



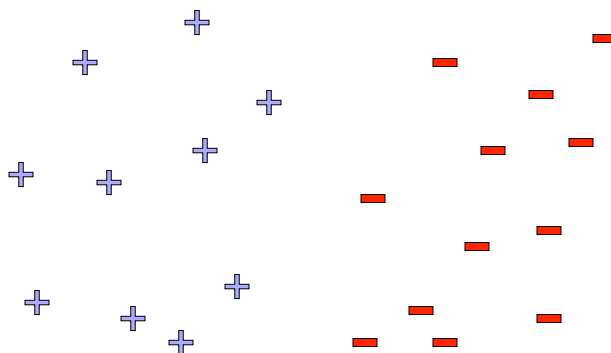
- Regularized least-squares (a.k.a. ridge regression), for $\lambda > 0$:

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \sum_j \left(t(\mathbf{x}_j) - \sum_i w_i h_i(\mathbf{x}_j) \right)^2 + \lambda \sum_{i=1}^k w_i^2$$

©Carlos Guestrin 2005-2014

17

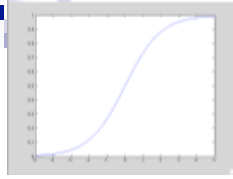
Linear Separability



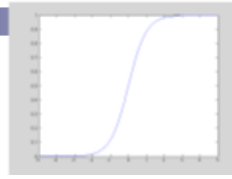
©Carlos Guestrin 2005-2014

18

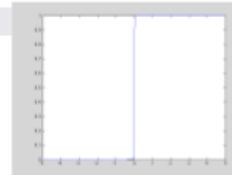
Large parameters → Overfitting



$$\frac{1}{1 + e^{-x}}$$



$$\frac{1}{1 + e^{-2x}}$$



$$\frac{1}{1 + e^{-100x}}$$

- If data is linearly separable, weights go to infinity
- In general, leads to overfitting:
- Penalizing high weights can prevent overfitting...

©Carlos Guestrin 2005-2014

19

Regularized Conditional Log Likelihood

- Add regularization penalty, e.g., L_2 :

$$\ell(\mathbf{w}) = \ln \prod_{j=1}^N P(y^j | \mathbf{x}^j, \mathbf{w}) - \frac{\lambda}{2} \|\mathbf{w}\|_2^2$$

- Practical note about w_0 :
- Gradient of regularized likelihood:

©Carlos Guestrin 2005-2014

20

Standard v. Regularized Updates

- Maximum conditional likelihood estimate

$$\mathbf{w}^* = \arg \max_{\mathbf{w}} \ln \prod_{j=1}^N P(y^j | \mathbf{x}^j, \mathbf{w})$$

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 | \mathbf{x}^j, \mathbf{w}^{(t)})]$$

- Regularized maximum conditional likelihood estimate

$$\mathbf{w}^* = \arg \max_{\mathbf{w}} \ln \prod_{j=1}^N P(y^j | \mathbf{x}^j, \mathbf{w}) - \frac{\lambda}{2} \sum_{i=1}^k w_i^2$$

$$w_i^{(t+1)} \leftarrow w_i^{(t)} + \eta \left\{ -\lambda w_i^{(t)} + \sum_j x_i^j [y^j - \hat{P}(Y^j = 1 | \mathbf{x}^j, \mathbf{w}^{(t)})] \right\}$$

©Carlos Guestrin 2005-2014

21

Please Stop!! Stopping criterion

$$\ell(\mathbf{w}) = \ln \prod_j P(y^j | \mathbf{x}^j, \mathbf{w}) - \lambda \|\mathbf{w}\|_2^2$$

- When do we stop doing gradient descent?

- Because $\ell(\mathbf{w})$ is strongly concave:
 - i.e., because of some technical condition

$$\ell(\mathbf{w}^*) - \ell(\mathbf{w}) \leq \frac{1}{2\lambda} \|\nabla \ell(\mathbf{w})\|_2^2$$

- Thus, stop when:

©Carlos Guestrin 2005-2014

22

Digression: Logistic regression for more than 2 classes

- Logistic regression in more general case (C classes), where Y in $\{0, \dots, C-1\}$

Digression: Logistic regression more generally

- Logistic regression in more general case, where Y in $\{0, \dots, C-1\}$

for $c > 0$

$$P(Y = c | \mathbf{x}, \mathbf{w}) = \frac{\exp(w_{c0} + \sum_{i=1}^k w_{ci}x_i)}{1 + \sum_{c'=1}^{C-1} \exp(w_{c'0} + \sum_{i=1}^k w_{c'i}x_i)}$$

for $c=0$ (normalization, so no weights for this class)

$$P(Y = 0 | \mathbf{x}, \mathbf{w}) = \frac{1}{1 + \sum_{c'=1}^{C-1} \exp(w_{c'0} + \sum_{i=1}^k w_{c'i}x_i)}$$

Learning procedure is basically the same as what we derived!